

CRM APPLICATION FOR JEWEL MANAGEMENT

College Name: Sri Krishna Adithya College of Arts and Science.

College Code: bruag

TEAM ID: NM2025TMID22133

TEAM MEMBERS:

Team Leader Name: Vijayashree S

Email: 23bsit165vijayashrees@skacas.ac.in

Team member1: Uthara Sekhar R

Email: 23bsit163utharasekharr@skacas.ac.in

Team member2: Vignesh S

Email: 23bsit164vigneshs@skacas.ac.in

Team member3: Vijaya Sri M

Email: 23bsit166vijayasrim@skacas.ac.in

1. INTRODUCTION

1.1 Project Overview

The jewellery industry is a cornerstone of the global economy, driven by consumer demand for luxury goods, personal adornment, and significant life event purchases. The market is highly competitive and dynamic, requiring businesses to manage not only valuable physical inventory but also intricate customer relationships. Traditional methods, relying on manual ledgers and rudimentary spreadsheets, often lead to significant operational bottlenecks, including data entry errors, inefficient inventory tracking, and a fragmented view of customer interactions. This results in lost sales opportunities, reduced customer satisfaction, and an inability to scale operations effectively.

The **CRM Application for Jewel Management** is a robust and scalable Salesforce-based solution designed to address these challenges head-on. By leveraging the power of Salesforce's cloud platform, this application provides a unified system to manage the entire customer lifecycle, from initial inquiry to after-sales service. It integrates core business functions such as customer relationship management, sales order processing, inventory control, and post-purchase support, all within a single, streamlined interface. The system employs key Salesforce features like custom objects, automated workflows (Flows), Apex triggers, and real-time dashboards to transform manual processes into efficient, data-driven operations. This project aims to empower jewellers with the tools necessary to enhance their operational efficiency, foster stronger customer loyalty, and gain a competitive edge in the market.



Flowchart



The flowchart for the CRM Application for Jewel Management provides a high-level overview of the system's architecture and its primary functions. It is designed to be a simple, illustrative diagram that helps stakeholders quickly understand what the application does.

At its core, the JEWEL MANAGEMENT CRM acts as the central hub for all business operations. It is a single, unified platform where data is stored, processes are automated, and key activities are managed.

Each of the five surrounding boxes represents a major module of the CRM:

1. **CUSTOMER MANAGEMENT:** This branch shows that the CRM is the central point for managing all customer data, from contact information to purchase history and preferences.
2. **SALES & BILLING:** This illustrates the CRM's role in streamlining the sales process, from processing new orders to automatically generating invoices and managing payments.
3. **INVENTORY TRACKING:** This highlights how the system provides real-time visibility into the stock of jewels, helping to prevent shortages and manage valuable assets efficiently.
4. **REPAIR REQUESTS:** This demonstrates the CRM's capability to track and manage all after-sales service and repair requests, ensuring a smooth and transparent process for customers.
5. **LOYALTY PROGRAMS:** This shows the system's function in automatically managing loyalty points and membership tiers, which is crucial for retaining customers and encouraging repeat business.

1.2 Purpose

The primary objective of this project is to provide a comprehensive and integrated solution for jewel businesses to manage their operations efficiently. Specifically, the system is designed to achieve the following:

- **Centralize customer information:** Create a single source of truth for all customer data, including contact details, purchase history, preferences, and loyalty status, enabling personalized service.
- **Automate business processes:** Eliminate manual data entry and human error by automating tasks such as invoice generation, payment reminders, and status updates through Salesforce Flows and email alerts.
- **Enhance inventory control:** Provide real-time visibility into jewel stock levels, track items by category (e.g., gold, silver, diamonds), and manage stock movements to prevent shortages or overstocking.
- **Streamline after-sales service:** Efficiently track and manage repair and service requests, ensuring timely resolution and clear communication with customers.
- **Boost customer loyalty:** Implement a structured loyalty program with membership tiers and points, encouraging repeat business and rewarding high-value customers.
- **Enable data-driven decision-making:** Offer managers real-time dashboards and reports on sales performance, inventory turnover, and repair service metrics, facilitating informed strategic decisions.



2. LITERATURE REVIEW

2.1 Existing System

The current landscape of jewel management is dominated by fragmented and outdated systems. Many small to medium-sized jewel businesses still rely on analogy methods such as handwritten registers and simple spreadsheets (e.g., Microsoft Excel) to record sales, track inventory, and manage customer contacts. While these tools are accessible and require minimal initial investment, they are highly inefficient and prone to error. Data duplication is common, and retrieving specific customer or sales information can be a time-consuming and cumbersome process. The lack of a centralized database makes it difficult to generate meaningful reports or analyse sales trends, severely limiting the business's ability to grow and adapt.



2.2 Limitations of Existing System

The limitations of manual and spreadsheet-based systems are manifold and significantly impact business performance:

- **Time-consuming and prone to human error:** Manual data entry is a slow process and is susceptible to typos and inconsistencies, leading to inaccurate records.
- **Lack of centralized customer data:** Customer information is siloed across multiple files or notebooks, making it impossible to get a 360-degree view of the customer relationship.
- **No automation for billing or reminders:** Invoices and payment reminders must be created and sent manually, often resulting in delayed payments and increased administrative overhead.
- **Poor tracking of repairs and loyalty programs:** Tracking a jewel through its repair journey is a manual, paper-based process, and loyalty points or tiers are difficult to manage without an automated system.
- **Inadequate reporting and analytics:** Generating reports on sales, inventory, or customer behavior is a labor-intensive process, and the data is often outdated by the time it is compiled.

2.3 Proposed Solution

To overcome these limitations, a modern, cloud-based Customer Relationship Management (CRM) system is proposed. The **Salesforce CRM platform** is the ideal solution due to its proven scalability, security, and extensive customization capabilities. Unlike generic software, Salesforce allows for the creation of custom objects and fields that are specific to the needs of the jewelry industry.

The proposed Salesforce CRM offers:

- **Cloud-based access anywhere, anytime:** Managers and sales representatives can access critical business data from any device, ensuring business continuity and flexibility.
- **Customizable objects:** Create bespoke objects (e.g., Customer, Product, Order, Invoice, Repair, Loyalty) to precisely model the business's data and processes.
- **Automated workflows:** Use native Salesforce automation tools like **Flows**, **Validation Rules**, and **Apex Triggers** to streamline operations and ensure data integrity.
- **Real-time dashboards for managers:** Provide visual, real-time insights into key performance indicators, such as sales trends, inventory status, and repair volumes, enabling agile decision-making.

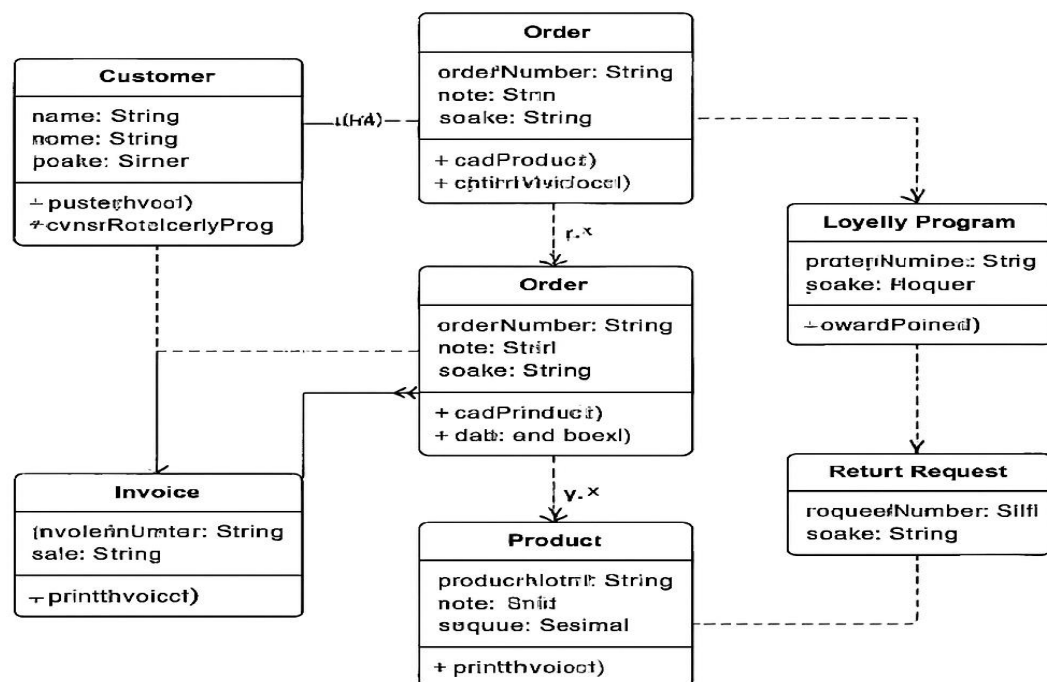
- **Enhanced customer satisfaction:** The centralized customer data and automated loyalty programs will enable a personalized customer experience, fostering stronger relationships and repeat business.



3. PROPOSED SYSTEM

The proposed Salesforce-based CRM application is a modular system built around the key functions of a jewel retail business. The system is designed to be intuitive for users while providing powerful backend automation. The main modules include:

- **Customer Management:** A comprehensive module to store customer details, track their purchase history, and manage their loyalty status.
- **Product Management:** A complete catalog of jewels and other products, including detailed information such as material (gold, silver, platinum), gemstone type, weight, price, and real-time stock availability.
- **Order & Invoice Management:** A seamless process for creating new orders, auto-generating invoices upon confirmation, and sending automated payment reminders to customers.
- **Repair Request Tracking:** A dedicated module to manage incoming repair and service requests, track their status (e.g., "Pending," "In Progress," "Completed"), and ensure timely customer communication.
- **Loyalty Program:** An automated system to assign loyalty points to customers based on their purchases and upgrade their membership tiers (e.g., "Silver," "Gold," "Diamond") based on predefined criteria.
- **Approval Process:** A critical business process for managing high-value transactions, where a manager's approval is required before an order can be finalized.



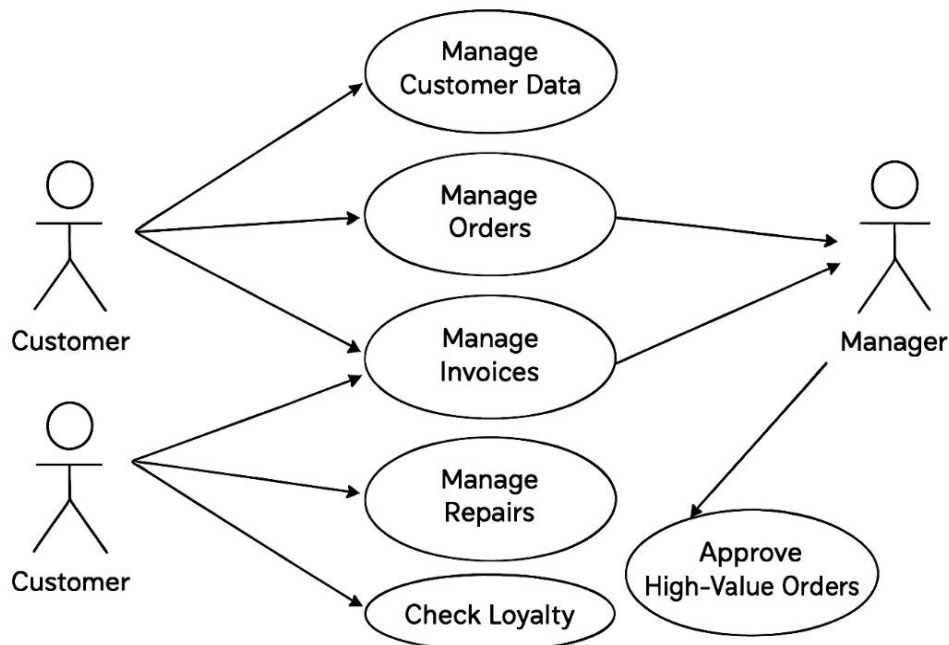
4. SYSTEM DESIGN

4.1 Use Case Diagram

The Use Case Diagram visually represents the interactions between different users (actors) and the system. The main actors are the **Customer**, **Sales Representative**, and **Manager**.

- **Customer:** Can place orders, request repairs, and check their loyalty points.
- **Sales Representative:** Has a more comprehensive role, managing customer data, processing new orders, issuing invoices, and creating repair requests.
- **Manager:** Possesses administrative privileges, including approving high-value orders and viewing reports and dashboards.

Use Case Diagram for Jewelry Salesforce CRM

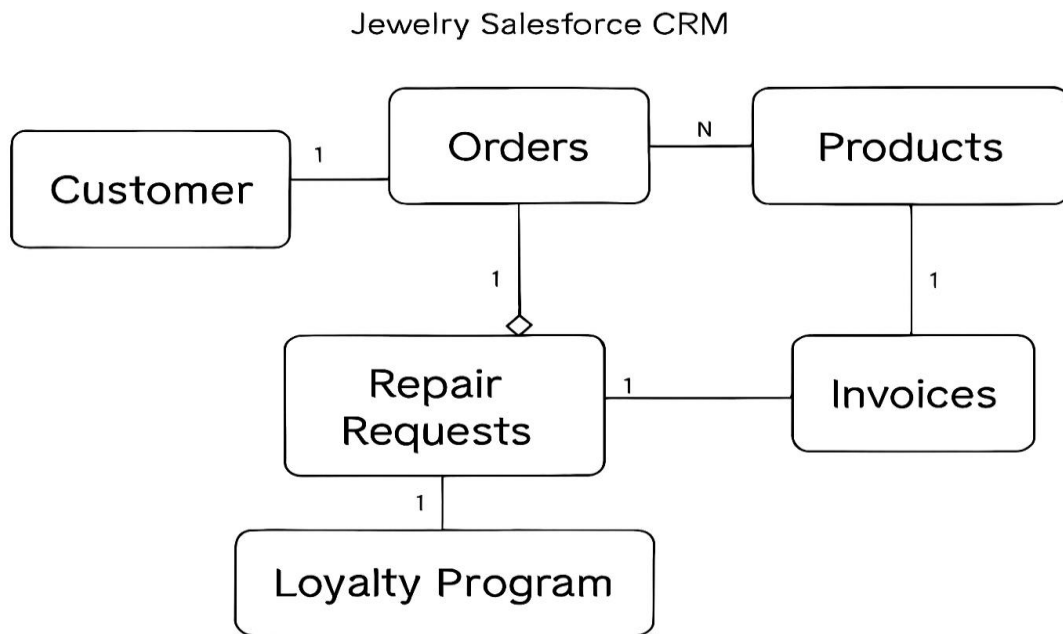


4.2 Entity Relationship Diagram (ERD)

The Entity Relationship Diagram (ERD) is crucial for defining the data model of the application. It illustrates how the custom objects in Salesforce relate to one another. The core relationships are as follows:

- **Customer ↔ Orders:** A customer can have multiple orders, and each order is associated with a single customer.
- **Orders ↔ Products:** An order can contain multiple products, and a product can be part of many orders. This is a many-to-many relationship often handled with a junction object (e.g., OrderProduct__c).
- **Orders ↔ Invoices:** An invoice is directly linked to an order.

- **Customer ↔ Repair Requests:** A customer can submit multiple repair requests, and each request is for a specific customer.
- **Customer ↔ Loyalty Program:** A customer record is linked to their loyalty points and tier status.

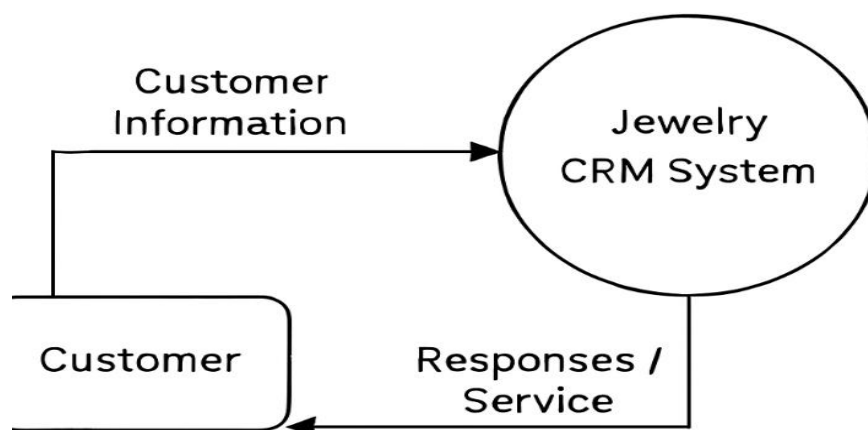


4.3 Data Flow Diagram (DFD)

The Data Flow Diagram (DFD) provides a high-level view of how data moves through the system.

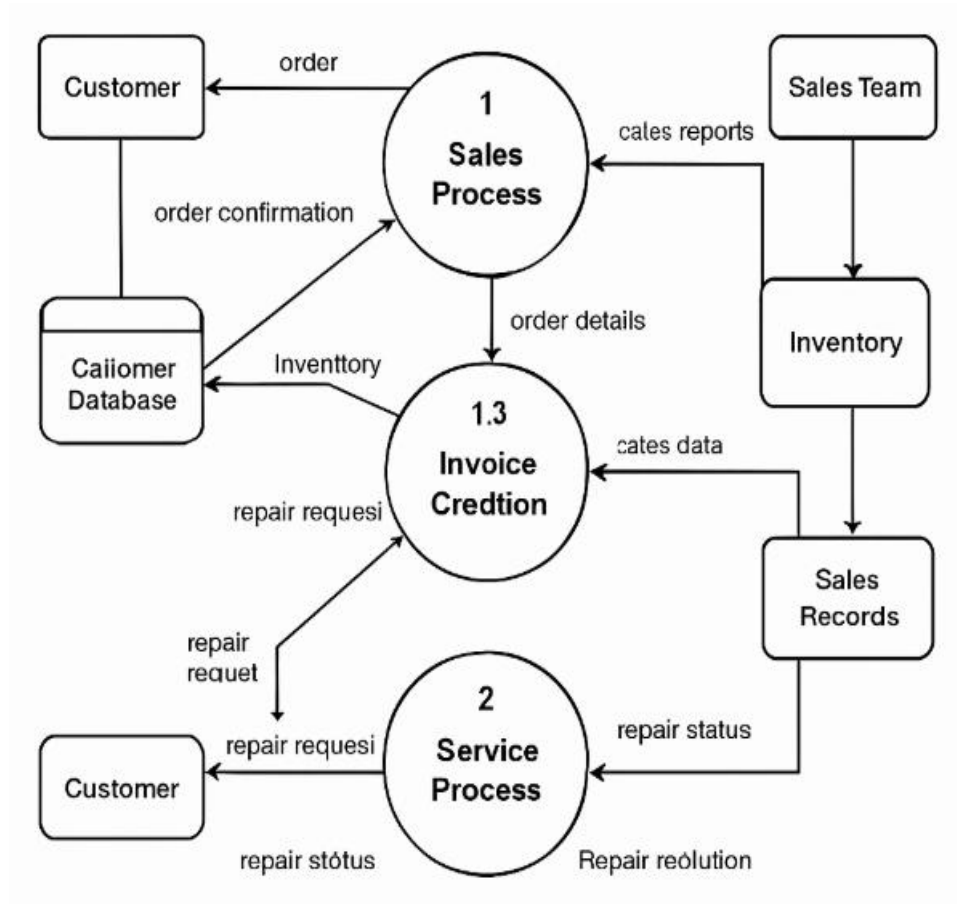
- **Level 0 (Context Diagram):** The customer interacts with the CRM system as a single process.

Data Flow Diagram Level 0 (Context Diagram)



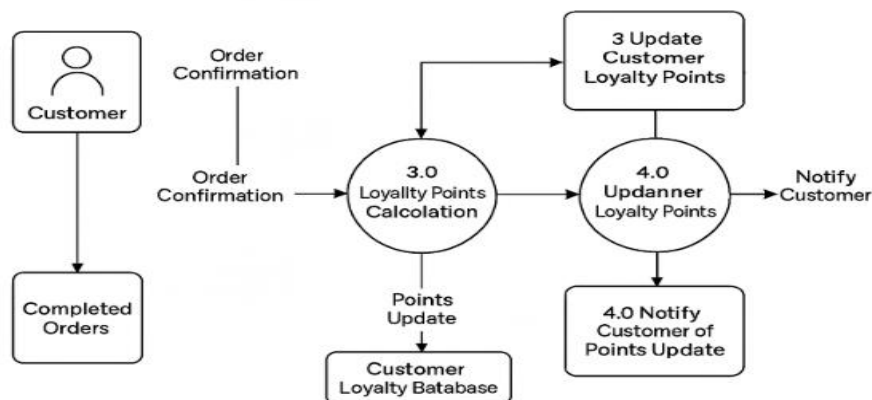
- **Level 1 (Main Processes):**

- **Sales Process:** A customer's order is processed, leading to the creation of an invoice.
- **Service Process:** A customer submits a repair request, which is handled and resolved by the service team.



- **Level 2 (Detailed Processes):**

- **Loyalty Process:** After an order is completed, a subprocess is triggered to calculate and allocate loyalty points, which then update the customer's record.

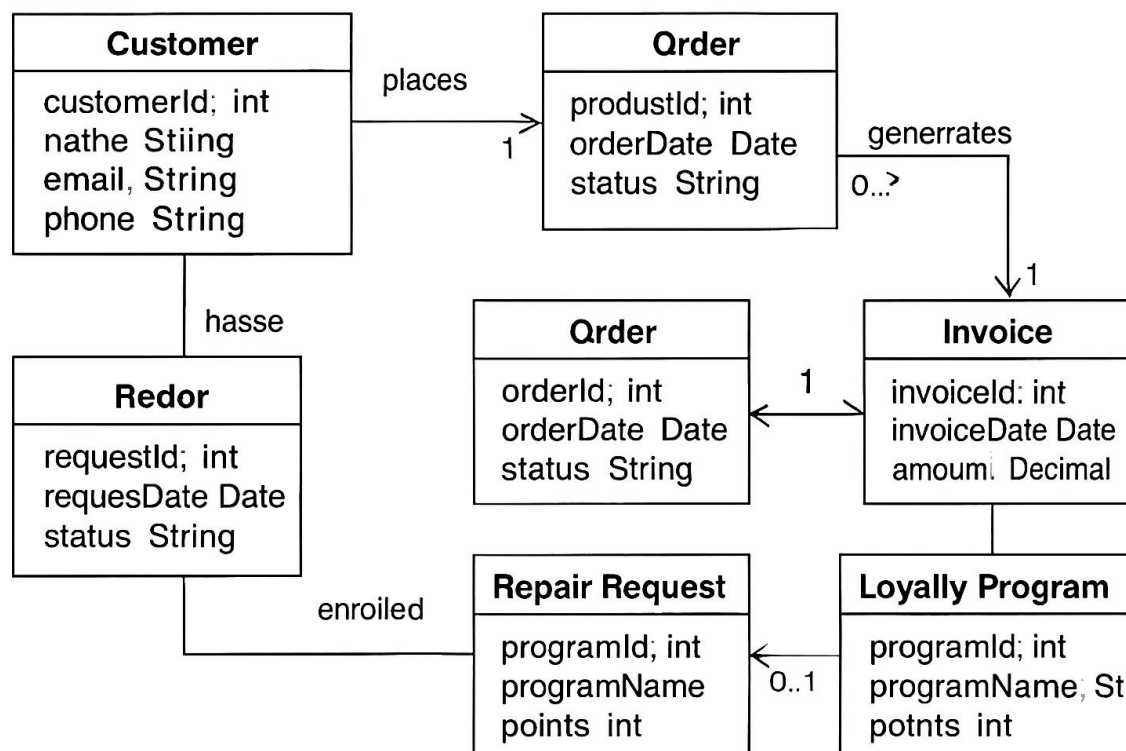


Jewelry CRM Loyalty Process. Level Loyalty Process

4.4 UML Diagrams

UML (Unified Modeling Language) diagrams provide a more detailed, object-oriented view of the system.

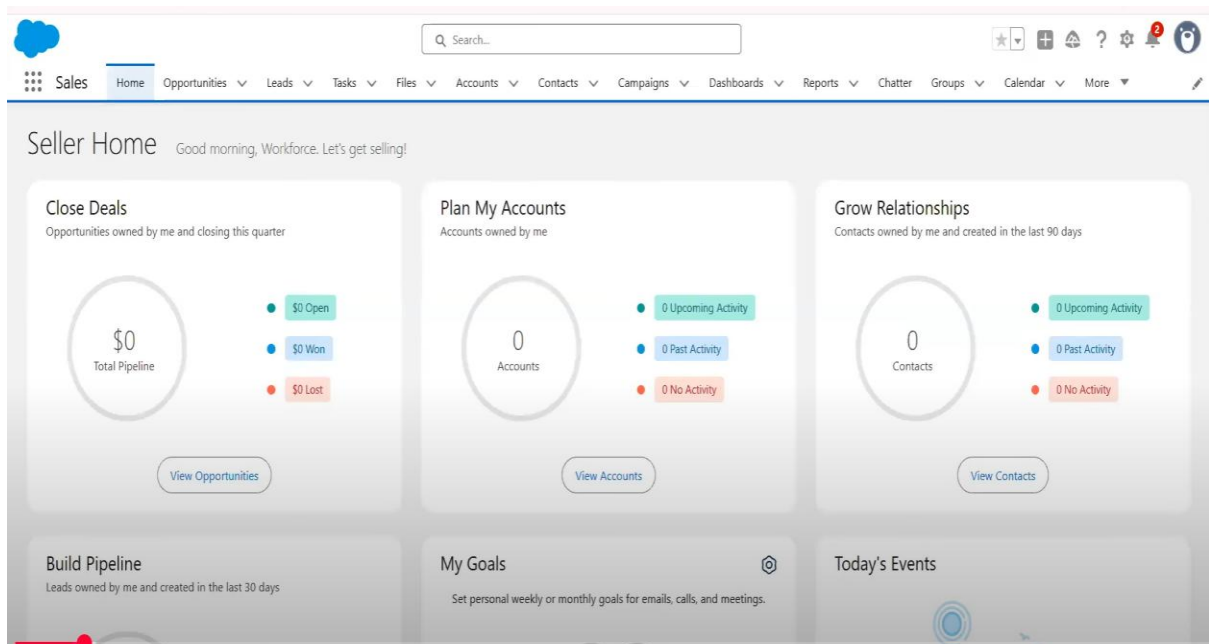
- **Class Diagram:** Shows the static structure of the system, including classes (custom objects like Customer__c, Product__c, Order__c, etc.), their attributes (fields), and their relationships.
- **Sequence Diagram:** Illustrates the interactions between objects in a time-ordered sequence. For example, a sequence diagram could show the flow when a customer places an order: Customer object interacts with Order object, which then calls the Invoice object to be generated.
- **Activity Diagram:** Models the flow of control and activities within a process, such as the steps involved in handling a repair request from submission to completion.



5. IMPLEMENTATION

- **Step 1: Salesforce Developer Account**

The first step in development is to create a free Salesforce Developer account. This environment provides all the tools and resources needed to build and test the application without any cost.



Sign up at [Salesforce Developer Signup](#).

- **Step 2: Object Creation**

Custom objects are the foundation of the application. The following objects were created with specific fields to capture jewel business data:

1. **Customer:** Fields for Name, Email, Phone, and Loyalty ID.
2. **Product:** Fields for Name, Category (e.g., Ring, Necklace), Weight, Price, and Stock.
3. **Order:** Fields for Order Date, Customer (Lookup), Product (Lookup), and Quantity.
4. **Invoice:** Fields for Invoice Number, Order Reference (Lookup), Amount, and Status (e.g., "Pending," "Paid").
5. **Repair Request:** Fields for Repair ID, Customer (Lookup), Item, and Status (e.g., "Received," "Repairing," "Completed").
6. **Loyalty Program:** Fields for Customer (Lookup), Points, and Tier.

[Setup](#)
[Home](#)
[Object Manager](#)

Object Manager
53+ Items, Sorted by Label

[Schema Builder](#)
[Create](#)

LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED
Account	Account	Standard Object			
Activity	Activity	Standard Object			
Address	Address	Standard Object			
Agent Work	AgentWork	Standard Object			
Alternative Payment Method	AlternativePaymentMethod	Standard Object			
API Anomaly Event Store	ApiAnomalyEventStore	Standard Object			
Appointment Category	AppointmentCategory	Standard Object			
Appointment Invitation	AppointmentInvitation	Standard Object			
Appointment Invitee	AppointmentInvitee	Standard Object			
Appointment Topic Time Slot	AppointmentTopicTimeSlot	Standard Object			

New Custom Object

Context-Sensitive Help Setting
☒ Open the standard Salesforce.com Help & Training window
☐ Open a window using a Visualforce page

Content Name

Enter Record Name Label and Format

The Record Name appears in page layouts, key lists, related lists, lookups, and search results. For example, the Record Name for Account is "Account Name" and for Case it is "Case Name".

Record Name
Example: **Account Name**

Data Type
Warning: If you plan to insert a high volume of records in this object, via the API for example, use a data type that supports large volumes of data.

Optional Features

- ☐ Allow Reports
- ☐ Allow Activities
- ☐ Track Field History
- ☐ Allow in Chatter Groups

New Custom Object Tab

Step 1. Enter the Details

Choose the custom object for this new custom tab. Fill in other details.

New Custom Object Tab

Select an existing custom object or [create a new custom object now](#).

Object

Tab Style

(Optional) Choose a Home Page Custom Link to show as a splash page the first time your users click on this tab.

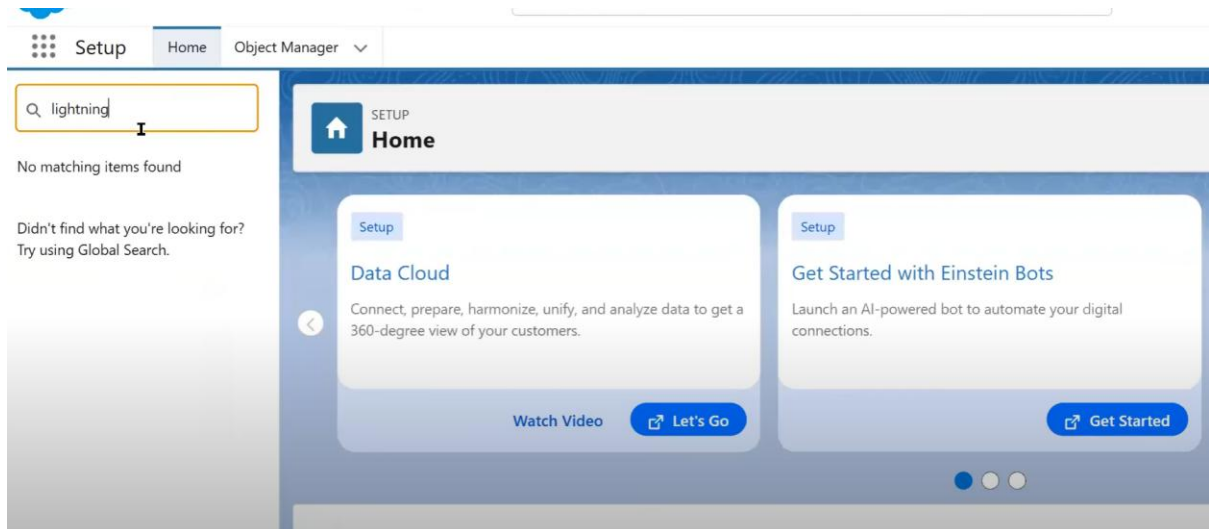
Splash Page Custom Link

Enter a short description.

Description

- **Step 3: Configurations**

After creating the objects, relationships were established to connect the data, and the user interface was configured for usability.



1. **Relationships:** Master-detail and lookup relationships were configured between Orders, Customers, and Products.
2. **Lightning App:** A custom Lightning App named "Jewel Management" was created with tabs for each of the core objects (Customers, Products, Orders, Invoices, etc.).

New Lightning App

Give your Lightning app a name and description. Upload an image and choose the highlight color for its navigation bar.

App Details

* App Name ⓘ

* Developer Name ⓘ

Description ⓘ

App Branding

Image ⓘ

Upload

Primary Color Hex Value ⓘ

#0070D2

Org Theme Options

☐ Use the app's image and color instead of the org's custom theme

SETUP > OBJECT MANAGER

Item

Details

- Fields & Relationships**
- Page Layouts
- Lightning Record Pages
- Buttons, Links, and Actions
- Compact Layouts
- Field Sets
- Object Limits
- Record Types
- Related Lookup Filters

Step 2. Enter the details

Field Label:

Values: ☐ Use global picklist value set ☒ Enter values, with each value separated by a new line

☐ Display values alphabetically, not in the order entered

☐ Use first value as default value

- Validation Rules:** Rules were implemented to ensure data quality, such as validating email formats, phone numbers, and preventing sales if stock is unavailable.

SETUP > OBJECT MANAGER

Jewel Customer

Record Types

Related Lookup Filters

Search Layouts

List View Button Layout

Restriction Rules

Scoping Rules

Object Access

Triggers

Flow Triggers

Validation Rules

Fields & Relationships

8 Items, Sorted by Field Label

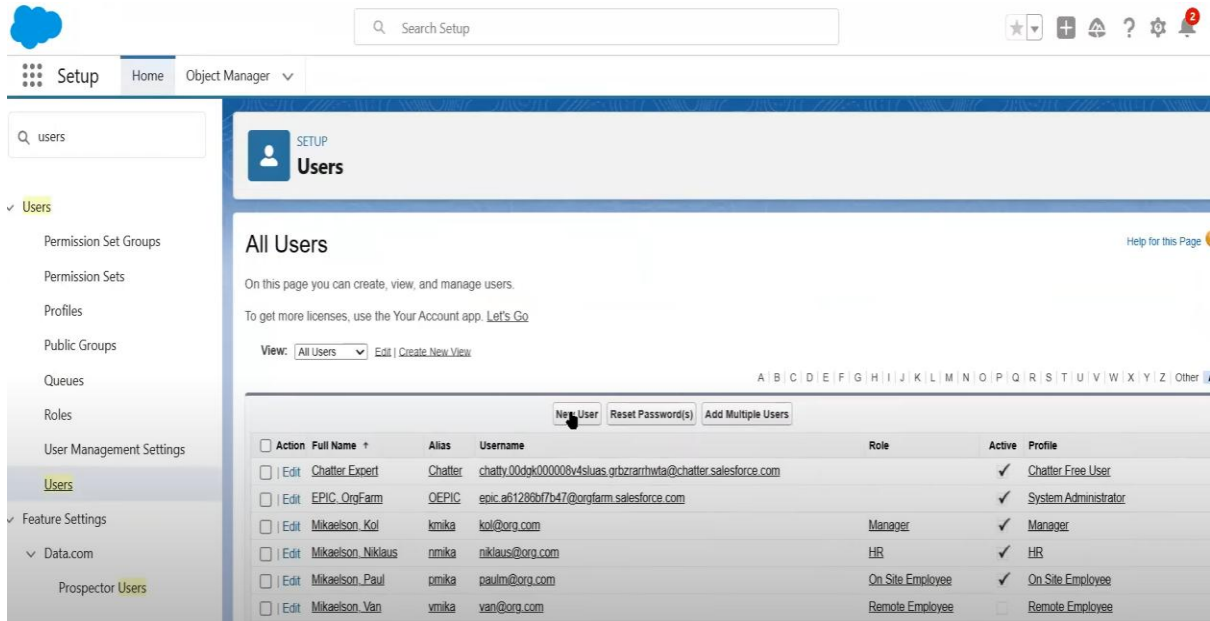
Field Label	Field Name	Field Type
City	City__c	Text(18)
Created By	CreatedById	Lookup(User)
Customer name	Name	Text(80)
Email	Email__c	Email
Last Modified By	LastModifiedById	Lookup(User)
Owner	OwnerId	Lookup(User,Group)
Phone	Phone__c	Text(18)

- Step 4: Automation**

Automating business logic is where Salesforce truly adds value. The following automations were implemented:

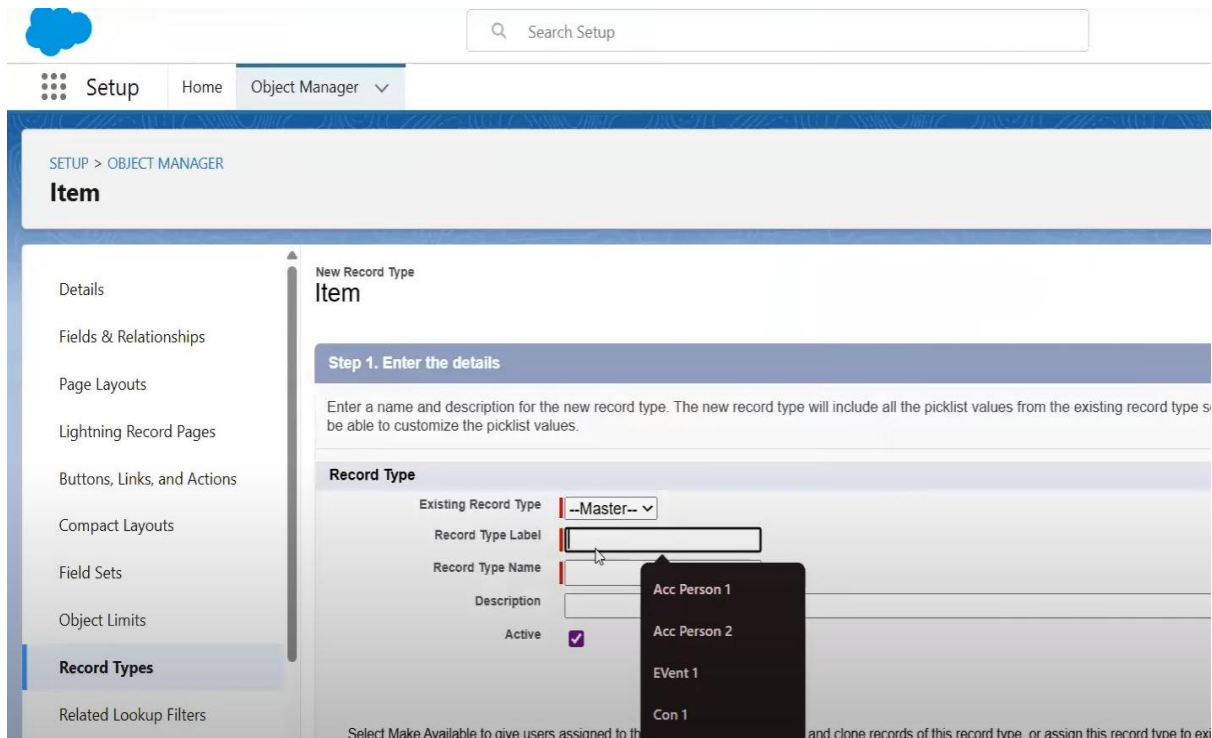
- Flows:** A Record-Triggered Flow was built to automatically generate an invoice record once an order's status is confirmed.

2. **Triggers:** An Apex Trigger (LoyaltyPointsTrigger) was developed to automatically add loyalty points to a customer's record after a successful purchase.
3. **Email Templates:** A set of email templates was created for purchase confirmations, payment reminders, and loyalty program updates.
4. **Scheduler:** An Apex Scheduled Class (PaymentReminderScheduler) was configured to run monthly, sending reminders for pending payments.
5. **Approval Process:** A multi-step approval process was set up to route orders above a certain value (e.g., ₹5,00,000) to a manager for mandatory approval.



The screenshot shows the Salesforce Setup page with the 'Users' section selected. The left sidebar lists various setup options, with 'Users' highlighted. The main content area displays the 'All Users' page, which includes a search bar, a 'View' dropdown set to 'All Users', and a table of users. The table has columns for Action, Full Name, Alias, Username, Role, Active, and Profile. A 'New User' button is visible above the table.

Action	Full Name	Alias	Username	Role	Active	Profile
<input type="checkbox"/> Edit	Chatter Expert	Chatter	chatty00dpg000006v4sluas.grbzrarrhwa@chatter.salesforce.com		✓	Chatter Free User
<input type="checkbox"/> Edit	EPIC_OrgFarm	OEPIE	epic.a61286b17b47@orgfarm.salesforce.com		✓	System Administrator
<input type="checkbox"/> Edit	Mikaelson_Kol	kmika	kol@org.com	Manager	✓	Manager
<input type="checkbox"/> Edit	Mikaelson_Niklaus	nmika	niklaus@org.com	HR	✓	HR
<input type="checkbox"/> Edit	Mikaelson_Paul	pmika	paulm@org.com	On Site Employee	✓	On Site Employee
<input type="checkbox"/> Edit	Mikaelson_Van	vmika	van@org.com	Remote Employee		Remote Employee



The screenshot shows the Salesforce Setup page with the 'Object Manager' section selected. The left sidebar lists various setup options, with 'Record Types' highlighted. The main content area displays the 'New Record Type' page for the 'Item' object. The page includes a 'Step 1. Enter the details' section with fields for Existing Record Type, Record Type Label, Record Type Name, and Description. A dropdown menu is open for the 'Record Type Name' field, showing options like 'Acc Person 1', 'Acc Person 2', 'Event 1', and 'Con 1'.

Step 1. Enter the details

Enter a name and description for the new record type. The new record type will include all the picklist values from the existing record type so be able to customize the picklist values.

Record Type

Existing Record Type: --Master--

Record Type Label: [Text Field]

Record Type Name: [Text Field]

Description: [Text Field]

Active: ☒

Select Make Available to give users assigned to the record type and clone records of this record type, or assign this record type to existing records.

Setup Home Object Manager

Search Setup

permission

Users

Permission Set Groups

Permission Sets

Custom Code

Custom Permissions

Didn't find what you're looking for? Try using Global Search.

Permission Sets

Permission Set Overview

Description	License	API Name	Per_to_Worker
Session Activation Required	<input type="checkbox"/>	Namespace Prefix	
Permission Set Groups Added To	0	Created By	Workforce Project, 8/25/2025, 12:04 AM
		Last Modified By	Workforce Project, 8/25/2025, 12:04 AM

Apps

- Assigned Apps**
Settings that specify which apps are visible in the app menu
- Assigned Connected Apps**
Settings that specify which connected apps are visible in the app menu
- Object Settings**
Permissions to access objects and fields, and settings such as tab availability
- App Permissions**
Permissions to perform app-specific actions, such as "Manage Call Centers"
- Apex Class Access**
Permissions to execute Apex classes
- Visualforce Page Access**
Permissions to execute Visualforce pages

Setup Home Object Manager

Search Setup

permission

Users

Permission Set Groups

Permission Sets

Custom Code

Custom Permissions

Didn't find what you're looking for? Try using Global Search.

Per to Worker

Select Users to Assign

All Users ▾

9 items • Sorted by Full Name • Filtered by All users • Updated a few seconds ago

Search this list...

<input type="checkbox"/>	Full Name ↑	AI...	Username	Role	A...	Profile
<input type="checkbox"/>	Integration User	integ	integration@00dggk000008v4sluas.com		<input checked="" type="checkbox"/>	Analytics Cloud Integration Us...
<input type="checkbox"/>	Kol Mikaelson	kmika	kol@org.com	Manager	<input checked="" type="checkbox"/>	Manager
<input type="checkbox"/>	Niklaus Mikaelson	nmika	niklaus@org.com	Gold Smith	<input checked="" type="checkbox"/>	Gold Smith
<input type="checkbox"/>	OrgFarm EPIC	OIEPIC	epica61286bf7b47@orgfarm.salesforce.com		<input checked="" type="checkbox"/>	System Administrator

New Apex class

Please enter a name for your new Apex class

UpdatePaidAmountTriggerHandler

OK Cancel

File Edit Debug Test Workspace Help < >

UpdatePaidAmountTriggerHandler.apxc *

Code Coverage: None API Version: 64

```
1 public class UpdatePaidAmountTriggerHandler {
2     public static void handleBeforeInsert(List<Billing__c> newBillings) {
3
4         for (Billing__c billing : newBillings) {
5
6             billing.Paid_Amount__c = billing.Paying_Amount__c;
7
8         }
9     }
10 }
11
12
13
14 public static void handleBeforeUpdate(Map<Id, Billing__c> oldBillingsMap, List<Billing__c> updatedBillings) {
15
16     for (Billing__c billing : updatedBillings) {
17
18         Billing__c oldBilling = oldBillingsMap.get(billing.Id);
19
20     }
21 }
```

Logs Tests Checkpoints Query Editor View State Progress Problems 5

Name	Line	Problem
UpdatePaidAmountTriggerHandler	6	Variable does not exist: Paying_Amount__c
UpdatePaidAmountTriggerHandler	6	Variable does not exist: Paid_Amount__c
UpdatePaidAmountTriggerHandler	20	Variable does not exist: Paid_Amount__c

orgfarm-a01ed0511a-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage

My Tabs My Tabs

File Edit Debug Test Workspace Help < >

UpdatePaidAmountTriggerHandler.apxc * UpdatePaidAmountTrigger.apxt *

Code Coverage: None API Version: 64

```
1 trigger UpdatePaidAmountTrigger on Billing__c (before insert, before) {
2     if (Trigger.isInsert) {
3
4         UpdatePaidAmountTriggerHandler.handleBeforeInsert(Trigger.new);
5
6     } else if (Trigger.isUpdate) {
7
8         UpdatePaidAmountTriggerHandler.handleBeforeUpdate(Trigger.oldMap, Trigger.new);
9
10    }
11 }
```

Jewellery Inventory ... Jewel Customers Items Prices Customer Orders Billings Reports Dashboards

Create Report

Category

Recently Used

All

Accounts & Contacts

Opportunities

Customer Support Reports

Leads

Campaigns

Activities

Contracts and Orders

Price Books, Products and Assets

Administrative Reports

Select a Report Type

Q price

Showing results for price

Report Type Name	Category
Price Books with Products	Standard
Items with Prices	Standard
Prices	Standard

Details

Prices

Standard Report Type

Start Report

Details

Fields (12)

Created By You

No Reports Yet

Created By Others

No Reports Yet

Objects Used in Report Type

My Tabs My Tabs All Bookmarks

Flow Builder

Select Elements Run Debug View Tests Save As New Version Save Activate

Auto-Layout

Record-Triggered Flow
Start
Object: **Billing** [Edit](#)
Trigger: **A record is created or updated**
Optimize for: **Actions and Related Records**
[+ Add Scheduled Paths \(Optional\)](#)
[Open Flow Trigger Explorer for Billing](#)

Run Immediately

End

Configure Start

☐ A record is updated
☒ A record is created or updated
☐ A record is deleted

Set Entry Conditions

Specify entry conditions to reduce the number of records that trigger the flow and the number of times the flow is executed. Minimizing unnecessary flow executions helps to conserve your org's resources.

If you create a flow that's triggered when a record is updated, we recommend first defining entry conditions. Then select the **Only when a record is updated to meet the condition requirements** option for When to Run the Flow for Updated Records.

Condition Requirements
None

Optimize Flow

Optimize the Flow for:

Fast Field Updates **Actions and Related Records**

My Tabs My Tabs All Bookmarks

ager

ow...

Description

* Body ⓘ

Insert a resource...

View as Rich Text

Hello
Customer Name:
Here are the details for the item you purchased with Jewellery Inventory System
Item Type: {!\$Record.Item__r.Item_Type__c}
Ornament:
Weight: grams
Amount:

Select an Option Select ... B I U

The "Item__r" field doesn't exist on the "Billing__c" object, or you don't have access to the field.

Flow Builder Billing Alert Flow - V1

Last saved on 8/25/2025, 12:59 PM Inactive Run Debug View Tests Save As New Version Save Activate

Free-Form

Errors and Warnings

Errors (1) Warnings (0)

You can't activate this automation until you resolve an error.

notice

- Provide at least one email recipient.

Start
Record-Triggered Flow
Object: **Billing** [Edit](#)
Trigger: **A record is created or updated**
Optimize for: **Actions and Related Records**
[+ Add Scheduled Paths \(Optional\)](#)
[Open Flow Trigger Explorer for Billing](#)

Run Immediately

Action
notice

6. TESTING & RESULTS

6.1 Functional Testing

A series of tests were conducted to validate the functionality of the system:

- **Validation Rules:** Tested by entering invalid data to ensure rules for email format, phone numbers, and stock availability worked as expected.
- **Flows:** Tested the order-to-invoice flow to confirm that invoices were generated accurately and automatically.
- **Triggers:** Tested the loyalty points trigger by completing an order and verifying that the correct number of points was credited to the customer's account.
- **Approval Processes:** Tested by creating a high-value order and confirming that it was routed to the manager for approval and that the status updated correctly.
- **Email Alerts:** Tested the email templates and alerts by confirming that notifications were sent for purchase confirmations, reminders, and approval requests.



6.2 Results

The implementation and testing confirmed the successful creation of a robust and efficient CRM application. The results demonstrate the project's effectiveness:

- **Efficient customer management:** The centralized database provides a single, easy-to-access view of all customer information.
- **Reduced manual effort:** Automated invoice generation and payment reminders have significantly reduced administrative workload.
- **Increased customer engagement:** The automated loyalty program has successfully engaged repeat customers and incentivized new purchases.
- **Improved business insights:** Real-time dashboards provide managers with immediate insights into business performance, enabling quicker and more informed decisions.



7. ADVANTAGES & DISADVANTAGES

Advantages

- **Automation:** Reduces manual data entry and human error, increasing operational efficiency.
- **Real-time data:** Provides up-to-the-minute information on sales, inventory, and customer activity.
- **Customer-centric approach:** Centralized data and loyalty programs lead to better customer service and stronger relationships.
- **Accessibility:** Cloud-based access allows for remote management and flexibility.
- **Scalability:** The Salesforce platform can easily scale to accommodate business growth and new features.

Disadvantages

- **Internet Dependency:** The system requires a stable internet connection to function.
- **Initial Training:** Users will need training to adapt from manual processes to the new Salesforce system.
- **Customization Efforts:** While Salesforce is highly customizable, initial setup and specific jewel-industry configurations require a dedicated development effort.



8. CONCLUSION

The CRM Application for Jewel Management is a transformative project that effectively integrates the complex operations of a jewel business with the powerful capabilities of Salesforce CRM. By automating key processes such as customer management, sales, billing, and loyalty programs, the system not only enhances efficiency and reduces manual work but also significantly improves customer satisfaction. The project serves as a compelling demonstration of how a flexible and customizable platform like Salesforce can be tailored to meet the unique needs of a specialized industry, paving the way for future growth and innovation.

- **Future Enhancements:**

1. **AI-powered sales forecasting:** Integrating Salesforce's Einstein AI to predict sales trends and optimize inventory.
2. **Chatbot integration:** Implementing a chatbot on the website or within the app for instant customer support.
3. **Mobile application:** Developing a custom mobile app for on-the-go access for sales representatives and managers.
4. **E-commerce integration:** Connecting the CRM directly to an online store to synchronize customer and order data seamlessly.

Jewelry Salesforce CRM

CONCLUSION SUMMARY



✓ Repressed Customer
Bulregætion



✓ Enfierood Sales
Prosiom



✓ Delus Driven
Ineghte

FUTURE EMEARGEMENTS



Asi Drivesed
Sutb: Renesaiting



Chaibod Integration



Mobile App
Reetrappnenit



E Commence
Inlegration

9. APPENDIX

- **Sample Apex Trigger (Loyalty Points)**

Apex

```
trigger LoyaltyPointsTrigger on Order__c (after insert) {  
    // Logic to execute after an Order record is inserted  
    for (Order__c ord : Trigger.new) {  
        if (ord.Status__c == 'Confirmed') {  
            Loyalty__c loyalty = new Loyalty__c(  
                Customer__c = ord.Customer__c,  
                Points__c = ord.TotalAmount__c / 1000  
            );  
            insert loyalty;  
        }  
    }  
}
```

- **Sample Scheduler (Payment Reminder)**

Apex

```
global class PaymentReminderScheduler implements Schedulable {  
    global void execute(SchedulableContext sc) {  
        // Retrieve all pending invoices  
        List<Invoice__c> invoices = [SELECT Id, Customer__r.Email__c, DueDate__c  
                                    FROM Invoice__c WHERE Status__c='Pending'];  
  
        // Send email reminders for each pending invoice  
        for (Invoice__c inv : invoices) {  
            Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();  
            email.setToAddresses(new String[]{inv.Customer__r.Email__c});  
            email.setSubject('Payment Reminder: Invoice Due');
```

```
        email.setPlainTextBody('Dear ' + inv.Customer__r.Name + ',\n\nYour payment for  
Invoice ID ' + inv.Id +  
  
        ' is pending. Please pay before ' + inv.DueDate__c + '.\n\nThank  
you,\nJewel Management Team');  
  
        Messaging.sendEmail(new Messaging.SingleEmailMessage[]{email});  
  
    }  
  
}  
  
}
```

- **Sample Email Template**

Subject: Thank You for Your Purchase

Body:

Dear [Customer Name],

Thank you for purchasing [Product Name]. Your invoice number is [Invoice ID].

Your loyalty points have been updated in your account.

We look forward to serving you again.